

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 月 3 1 日
Date of Application:

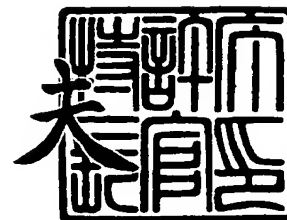
出 願 番 号 特 願 2 0 0 3 - 0 2 3 4 4 4
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 2 3 4 4 4]

出 願 人 ヤマハ株式会社
Applicant(s):

2 0 0 3 年 9 月 2 4 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 J99511A1

【提出日】 平成15年 1月31日

【あて先】 特許庁長官 殿

【国際特許分類】 G09G 5/00

【発明の名称】 画像処理装置

【請求項の数】 3

【発明者】

【住所又は居所】 静岡県浜松市中沢町 1 0 番 1 号 ヤマハ株式会社内

【氏名】 吉田 佳司

【特許出願人】

【識別番号】 000004075

【氏名又は名称】 ヤマハ株式会社

【代理人】

【識別番号】 100064908

【弁理士】

【氏名又は名称】 志賀 正武

【選任した代理人】

【識別番号】 100089037

【弁理士】

【氏名又は名称】 渡邊 隆

【手数料の表示】

【予納台帳番号】 008707

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9001626



【プルーフの要否】 要



【書類名】 明細書

【発明の名称】 画像処理装置

【特許請求の範囲】

【請求項 1】 圧縮された画像データをデコードして圧縮前の画像データに戻すデコーダと、

前記デコーダによってデコードされた画像データを第 1 の記憶手段に書き込む書込手段と、

前記第 1 の記憶手段から前記画像データを読み出す読出手段と、

前記読出手段によって読み出された画像データに所定の処理を行って第 2 の記憶手段に書き込む制御手段と、

前記第 2 の記憶手段から画像データを読み出し、表示装置へ出力する表示手段とを具備する画像処理装置において、

前記書込手段および前記読出手段が、前記第 1 の記憶手段をファーストイン・ファーストアウトメモリとして読出／書込制御することを特徴とする画像処理装置。

【請求項 2】 圧縮されたスプライトの画像データが記憶されたメモリと、前記スプライトの属性が記憶された属性テーブルと、

前記属性テーブル内のデータに基づいて前記メモリからスプライト画像データを読み出し、圧縮前のスプライト画像データに戻すデコーダと、

前記デコーダによってデコードされたスプライト画像データを第 1 の記憶手段に書き込む書込手段と、

前記第 1 の記憶手段から前記スプライト画像データを読み出す読出手段と、

前記読出手段によって読み出されたスプライト画像データに前記属性テーブル内のデータに基づく所定の処理を行って第 2 の記憶手段に書き込む制御手段と、

前記第 2 の記憶手段からスプライト画像データを読み出し、表示装置へ出力する表示手段とを具備する画像処理装置において、

前記書込手段および前記読出手段が、前記第 1 の記憶手段をファーストイン・ファーストアウトメモリとして読出／書込制御することを特徴とする画像処理装置。

【請求項 3】 前記制御手段における所定の処理は、画像データの拡大、縮小、回転、変形等のレンダリング処理であることを特徴とする請求項 1 または請求項 2 に記載の画像処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、描画性能の向上を図った画像処理装置に関する。

【0002】

【従来の技術】

パチンコやテレビゲーム等においては、スプライト（小画像）を使用した画像表示がしばしば行われる。図 7 はスプライトを使用した画像表示装置の構成を示すブロック図であり、この図において、符号 1 は CPU（中央処理装置）、2 は画像処理装置、3 は圧縮されたスプライトのパターンが記憶されたパターン ROM、4 は液晶等による表示装置である。画像処理装置 2 は CPU 1 からの指示を受け、パターン ROM 3 からスプライトパターンを読み出し、読み出したパターンをデコード（伸張）して圧縮前のスプライトパターンに戻し、内部のスプライトバッファに展開する。次いで、スプライトバッファからパターンを読み出し、拡大、縮小、回転、変形等のレンダリング処理を行って内部のフレームバッファに描画する。次いで、描画したパターンデータを表示装置 4 の水平／垂直走査タイミングに合わせて読み出し、表示装置 4 へ出力する。従来のスプライトによる表示処理に関する先行技術文献として特許文献 1～3 が知られている。

【0003】

【特許文献 1】

特開 2002-16810 号公報

【特許文献 2】

特開 2002-112263 号公報

【特許文献 3】

特開 2002-341859 号公報

【0004】

**【発明が解決しようとする課題】**

ところで、従来の画像処理装置においては、スプライトパターンをスプライトバッファに展開する処理と、展開されたパターンに対するレンダリング処理を行ってフレームバッファの所定位置に描画する処理を同期化して行っていたため、どちらか遅い方の処理がボトルネックとなり描画性能を低下させていた。

本発明は上記事情を考慮してなされたもので、その目的は、描画性能の向上を図った画像処理装置を提供することにある。

【0 0 0 5】**【課題を解決するための手段】**

この発明は上記の課題を解決するためになされたもので、請求項 1 に記載の発明は、圧縮された画像データをデコードして圧縮前の画像データに戻すデコーダと、前記デコーダによってデコードされた画像データを第 1 の記憶手段に書き込む書込手段と、前記第 1 の記憶手段から前記画像データを読み出す読出手段と、前記読出手段によって読み出された画像データに所定の処理を行って第 2 の記憶手段に書き込む制御手段と、前記第 2 の記憶手段から画像データを読み出し、表示装置へ出力する表示手段とを具備する画像処理装置において、前記書込手段および前記読出手段が、前記第 1 の記憶手段をファーストイン・ファーストアウトメモリとして読出／書込制御することを特徴とする画像処理装置である。

【0 0 0 6】

請求項 2 に記載の発明は、圧縮されたスプライトの画像データが記憶されたメモリと、前記スプライトの属性が記憶された属性テーブルと、前記属性テーブル内のデータに基づいて前記メモリからスプライト画像データを読み出し、圧縮前のスプライト画像データに戻すデコーダと、前記デコーダによってデコードされたスプライト画像データを第 1 の記憶手段に書き込む書込手段と、前記第 1 の記憶手段から前記スプライト画像データを読み出す読出手段と、前記読出手段によって読み出されたスプライト画像データに前記属性テーブル内のデータに基づく所定の処理を行って第 2 の記憶手段に書き込む制御手段と、前記第 2 の記憶手段からスプライト画像データを読み出し、表示装置へ出力する表示手段とを具備する画像処理装置において、前記書込手段および前記読出手段が、前記第 1 の記憶

手段をファーストイン・ファーストアウトメモリとして読出／書込制御すること
を特徴とする画像処理装置である。

【0007】

請求項3に記載の発明は、請求項2または請求項3に記載の画像処理装置において、前記制御手段における所定の処理は、画像データの拡大、縮小、回転、変形等のレンダリング処理であることを特徴とする。

【0008】

【発明の実施の形態】

以下、図面を参照し、この発明の実施の形態について説明する。図1はこの発明の一実施の形態による画像処理装置11の構成を示すブロック図である。この図において、符号13はスプライト属性テーブルであり、スプライト属性データがCPU（図示略）によって登録される。ここで、スプライト属性データには、パターンROM3内のスプライトパターンの格納アドレス、パターンの拡大、縮小、回転、変形等を定めるパラメータ、表示位置を指示するデータ等がある。14はデコードコントローラであり、パターンROM3内の圧縮されたスプライトパターンのデコード処理を制御する。15はROMインターフェイスであり、パターンROM3へ読出アドレスを出力し、パターンROM3から読み出されたパターンデータをデコーダ16へ出力する。

【0009】

デコーダ16はパターンROM3から読み出されたパターンデータのデコード（伸張処理）を行ってスプライトバッファインターフェイス17へ出力する。スプライトバッファインターフェイス17は、デコーダ16から出力されるスプライトパターンをスプライトバッファ18に展開する。また、レンダリングエンジン21からの指示を受けてスプライトバッファ18内のパターンデータを読み出し、レンダリングエンジン21へ出力する。ここで、スプライトバッファ18は複数のスプライトパターンを展開できる容量をもっている。20はレンダリングコントローラであり、スプライト属性テーブル13内のスプライト属性データを読み出し、読み出した属性データをレンダリングエンジン21へ出力する。

【0010】

レンダリングエンジン 21 は、レンダリングコントローラ 20 の指示に従ってスプライトバッファ 18 から読み出されたパターンデータのレンダリング処理を行い、その処理済みのデータをフレームバッファインターフェイス 22 へ出力する。フレームバッファインターフェイス 22 はレンダリングエンジン 21 から出力されるパターンデータをフレームバッファ 23 に描画する。また、ディスプレイコントローラ 25 からの指示を受けてフレームバッファ 23 からパターンデータを読み出し、ディスプレイコントローラ 25 へ出力する。フレームバッファ 23 は表示装置 4 の表示ドット対応でパターンデータが書き込まれるメモリであり、ダブルバッファ構成となっている。ディスプレイコントローラ 25 は画像表示のための各種のタイミング信号を生成して表示装置 4 へ出力し、また、フレームバッファ 23 から読み出されたパターンデータをそのタイミング信号に同期させて表示装置 4 へ出力する。

【0011】

次に、上述した画像処理装置 11 の動作を図 2 および図 3 に示すフローチャートを参照して説明する。

デコードコントローラ 14 は、まず、スプライト属性テーブル 13 にアクセスし、パターン ROM 3 のスプライト格納アドレスを取得する（図 2 のステップ S a 1）。次に、取得したアドレスを ROM インターフェイス 15 へ出力し、1 ブロックの読み出しを指示する。ここで、1 ブロックは 16×16 ドットのデータであり、スプライトパターンは通常 n （ n : 1 より大きい整数）ブロックのデータによって構成されている。上記の指示を受けた ROM インターフェイス 15 は、パターン ROM 3 から 1 ブロックのパターンデータを読み出し、デコーダ 16 へ出力する。デコーダ 16 はそのパターンデータのデコードを行う（ステップ S a 2）。そして、1 ブロックのデコードが終了すると（ステップ S a 3）、スプライトを構成する全ブロックのデコードが終了したか否かをチェックし（ステップ S a 4）、終了していない場合は、ROM インターフェイス 15 が次のブロックの読み出しを行い、読み出されたパターンデータがデコーダ 16 によってデコードされる（ステップ S a 2）。

【0012】

以下、上述した動作が繰り返えされ、最初のスプライトを構成する全ブロックのデコードが終了すると（ステップS a 4がYES）、再びステップS a 1へ戻り、次のスプライトの格納アドレスをスプライト属性テーブル13から読み出す。以後、上記と同様にして次のスプライトパターンのデコード処理が行われる。

【0013】

一方、デコーダ16において1ブロックのデコード処理が終了すると、スプライトバッファインターフェイス17によってデコード終了後のパターンデータがスプライトバッファ18に展開される（ステップS a 5）。ここで、スプライトバッファ18はFIFO（ファーストイン・ファーストアウト）メモリとして機能し、ブロック単位でパターンデータが順次展開される。スプライトバッファインターフェイス17は、まず、スプライトバッファ18がFULLの状態にあるか否かをチェックし（ステップS a 6）、否であった場合に展開を実行する（ステップS a 7）。

【0014】

そして、1ブロックの展開が終了すると（ステップS a 8）、全ブロックの展開が終了したか否かをチェックし、終了していない場合（ステップS a 9がNO）、再び、ステップS a 5へ戻り、次のブロックのデコードが終了するのを待って、そのブロックのパターンデータをスプライトバッファ18に展開する。また、全ブロックの展開が終了した場合は（ステップS a 9がYES）ステップS a 1へ戻る。

このように、スプライトバッファ18には、パターンROM3内のスプライトパターンがブロック単位でデコードされた後、FIFO方式で順次展開される。

【0015】

次に、フレームバッファ23の描画処理について図3を参照して説明する。

レンダリングコントローラ20は、スプライト属性テーブル13から、スプライトパターンの拡大、縮小、回転、変形等を定めるパラメータ、表示位置を指示するデータ等のスプライト属性データを読み出し（ステップS b 1）、レンダリングエンジン21へ出力する。レンダリングエンジン21は、レンダリングコントローラ20から受けたパラメータに基づいて初期パラメータの計算を行う（ス

テップ S b 2)。次に、スプライトバッファ 1 8 が E M P T Y であるか否かをチェックし（ステップ S b 3）、E M P T Y でない場合は、描画処理を行う（ステップ S b 4）。

【 0 0 1 6 】

すなわち、まず、スプライトバッファインターフェイス 1 7 へスプライトパターンの読み出しを指示する。スプライトバッファインターフェイス 1 7 は、その指示を受け、スプライトバッファ 1 8 からレンダリングエンジン 2 1 で属性データに基づいて計算された座標（アドレス）に該当するデータを読み出し、レンダリングエンジン 2 1 へ出力する。レンダリングエンジン 2 1 は、そのデータをレンダリングコントローラ 2 0 から受けた表示位置を示すデータと共にフレームバッファインターフェイス 2 2 へ出力する。フレームバッファインターフェイス 2 2 は、そのパターンデータを、表示位置データに対応するアドレスに描画する（ステップ S b 4、S b 5）。

【 0 0 1 7 】

フレームバッファ 2 3 への描画処理は、回転、変形等の処理があるため、スプライトバッファ 1 8 の E M P T Y の確認はスプライト単位で判断している。スプライト 1 個の全体のイメージがスプライトバッファ 1 8 に展開されていないと、描画処理を開始できない。そして、スプライトの全描画が終了すると（ステップ S b 6 が Y E S）、ステップ S b 1 へ戻り、レンダリングコントローラ 2 0 が次のスプライトの属性データをスプライト属性テーブル 1 3 から読み出す。そして、読み出した属性データに基づいて次のスプライトの描画処理が行われる。

【 0 0 1 8 】

このように、スプライトバッファ 1 8 の読み出しは、書き込みと非同期で行われる。すなわち、スプライトバッファ 1 8 は、前述したように複数のスプライトパターンを展開できる容量をもっており、そのスプライトバッファ 1 8 を 1 6 × 1 6 ドットを 1 ブロックとする単位で F I F O 化することにより、F I F O が F U L L にならない限りフレームバッファ 2 3 の描画終了を待たずに次のスプライトのパターンデータをデコードしてスプライトバッファ 1 8 に展開することができる。また、レンダリングコントローラ 2 0 は F I F O が E M P T Y でなければ

フレームバッファ 23 へ描画を継続することができる。これにより、パターン ROM 3 内のパターンデータのデコード処理とフレームバッファ 23 の描画処理の時間の違いを考慮することなくそれぞれの処理を行うことができ、この結果、描画処理能力を向上させることができる。

【0019】

次に、図 4～図 6 を参照しフレームバッファ 23 の書き込み／読み出し動作を詳述する。

フレームバッファ 23 に描画するためには、スプライトバッファ 18 からパターンデータを読み出す必要がある。スプライトバッファインターフェイス 17 では、デコーダ 16 におけるデコード処理を止めないように、スプライトバッファ 18 の展開アクセスを優先する。したがって、展開アクセスが行われていない時間においてスプライトバッファ 18 のパターンデータの読み出しが行われる。このため、スプライトバッファ 18 に展開している期間はフレームバッファ 23 の描画はできない。

【0020】

一方、フレームバッファインターフェイス 22 においては、表示装置 4 の画面にノイズを出さないため、フレームバッファ 23 の表示アクセス（すなわち、読み出しアクセス）が優先される。したがって、表示アクセス期間においては、フレームバッファ 23 の描画を行うことができない。

【0021】

このように、フレームバッファ 23 の描画は、スプライトバッファ 18 の展開アクセスの期間およびフレームバッファ 23 の表示アクセスの期間の双方において実行できず、このため、描画性能が悪い問題があった。そこで、この実施形態においては、フレームバッファ 23 の表示アクセス期間をスプライトバッファ 18 の展開アクセス期間に合わせる（同期させる）ことによって両アクセスを同じ期間において行うようにし、これにより、フレームバッファの描画性能の向上を図っている。

【0022】

以下、図 4～図 6 を参照して詳述する。図 4 はダブルバッファとして構成され

ているフレームバッファ 2 3 の表示／描画切換動作を示すタイミングチャートであり、図において V-BLANKN はフレーム周期のタイミングを示し、BANK0 はダブルバッファの一方を、BANK1 は他方を示している。この図に示すように、BANK0 と BANK1 は 1 フレームの表示が行われる毎に表示・描画・表示・・・と切り換えられる。なお、BANK0 と BANK1 のバスラインは共通であり、このため、ダブルバッファ構成であっても描画と表示（読出）を同時に行うことはできない。また、ダブルバッファ構成としているのは、描画期間を 1 フレームとるためであり、所定フレームの表示中においてその内容が書き換えられるのを防ぐためである。また、この実施形態はシングルフレームバッファ構成の場合も適用可能である。

【 0 0 2 3 】

図 5 はフレームバッファ 2 3 の表示アクセスのタイミングを説明するためのタイミングチャートであり、HDSP は 1 ラインの水平表示期間を示し、“ L ” レベルの期間が水平非表示期間を示している。表示アクセスは、表示装置 4 の画面の次のラインに表示するデータを現在のラインの表示期間の終了のタイミング（すなわち、水平非表示期間の開始タイミング）でディスプレイコントローラ 2 5 からフレームバッファインターフェイス 2 2 にリクエストされる（図 5 の DSPREQ）。1 ライン分の表示データは次のラインの表示期間の開始時点までにフレームバッファから読み出されることが必要である。一方、スプライトバッファ 1 8 の展開アクセスは、例外を除いて、水平非表示期間に最低 1 回は発生する。そこで、フレームバッファインターフェイス 2 2 は、ディスプレイコントローラ 2 5 からのリクエストを待機させておき、スプライトバッファ 1 8 の展開アクセスのタイミング（図 5 の OPENREQ）の立ち上がり時点から表示アクセス（図 5 の DSPACK）を開始する。

【 0 0 2 4 】

これにより、スプライトバッファ 1 8 の展開アクセスと、フレームバッファ 2 3 の表示アクセスを同期させることができる。スプライトバッファ 1 8 の展開アクセスの期間とフレームバッファ 2 3 の表示アクセスの期間がほぼ等しい場合、両者の終了タイミングもほぼ等しくなり、終了直後から滞りなくフレームバッファ 2 3 への描画を再開することが可能となる。

【0025】

他方、スプライトバッファ18の展開アクセスが発生しない状態で、水平表示期間の開始時点から逆算して1ライン分の表示データを読み出せるタイミングとなった場合、フレームバッファインターフェイス22はスプライトバッファ18の展開アクセスの有無にかかわらずフレームバッファ23の表示アクセスを強制的に実行する。図6はこの場合のタイミングチャートであり、DSPLIMITが表示アクセスの強制実行のタイミングを示している。これにより、例外的にスプライトバッファ18の展開アクセスがなかった場合も、間違いなくフレームバッファ23の表示アクセスを実行することができる。

【0026】

【発明の効果】

以上説明したように、この発明によれば、第1の記憶手段（スプライトバッファ）をファーストイン・ファーストアウトメモリとして読出／書込制御するようにしたので、圧縮された画像データのデコード処理と、第2の記憶手段（フレームバッファ）の書込処理の時間の違いを考慮することなくそれぞれの処理を行うことができ、この結果、描画処理能力を向上させることができる。

【図面の簡単な説明】

【図1】 この発明の一実施形態による画像処理装置の構成を示すブロック図である。

【図2】 同実施形態の動作を説明するためのフローチャートである。

【図3】 同実施形態の動作を説明するためのフローチャートである。

【図4】 同実施形態におけるフレームバッファ23の描画／表示タイミングを示すタイミングチャートである。

【図5】 同実施形態におけるフレームバッファ23の表示タイミングを説明するためのタイミングチャートである。

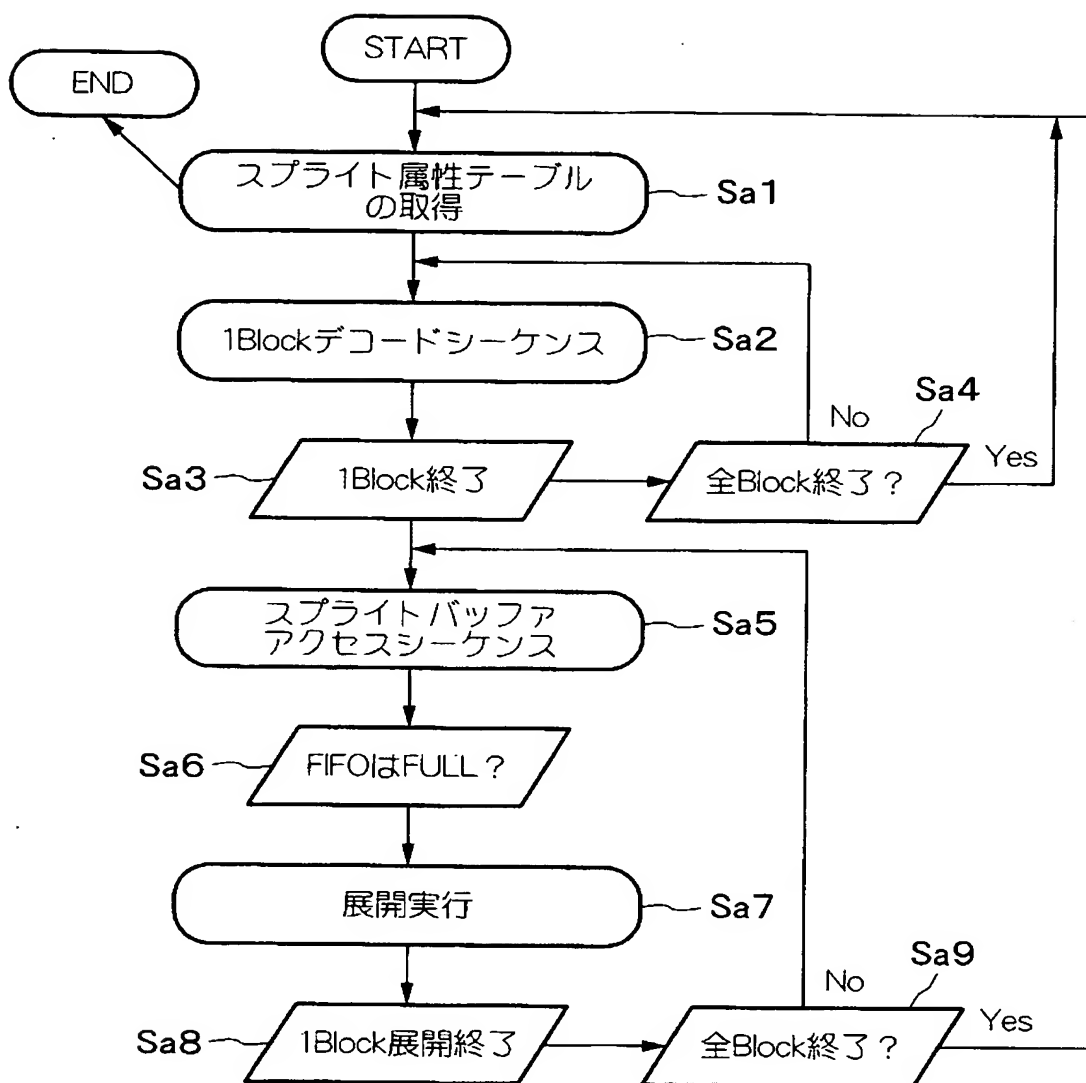
【図6】 同実施形態におけるフレームバッファ23の表示タイミングを説明するためのタイミングチャートである。

【図7】 スプライトを使用した画像表示装置の構成を示すブロック図である。

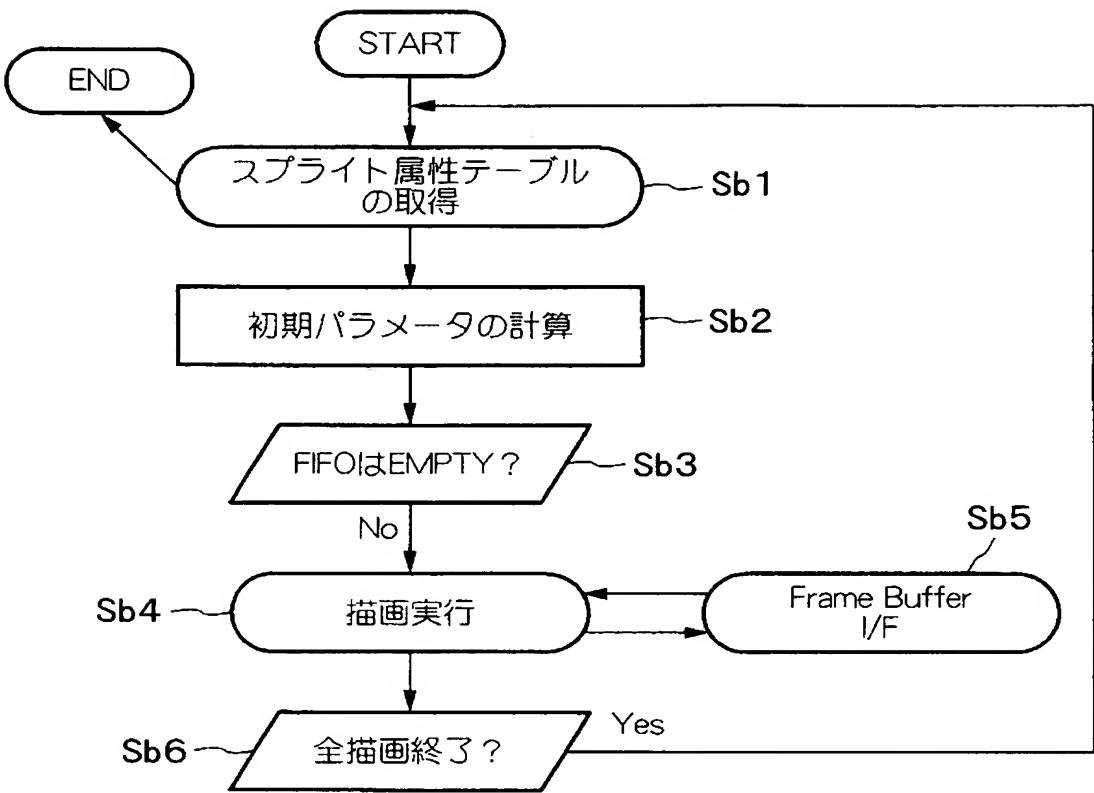
【符号の説明】

3…パターンROM、4…表示装置、11…画像処理装置、13…スプライト属性テーブル、14…デコードコントローラ、15…ROMインターフェイス、16…デコーダ、17…スプライトバッファインターフェイス、18…スプライトバッファ、20…レンダリングコントローラ、21…レンダリングエンジン、22…フレームバッファインターフェイス、23…フレームバッファ、25…ディスプレイコントローラ。

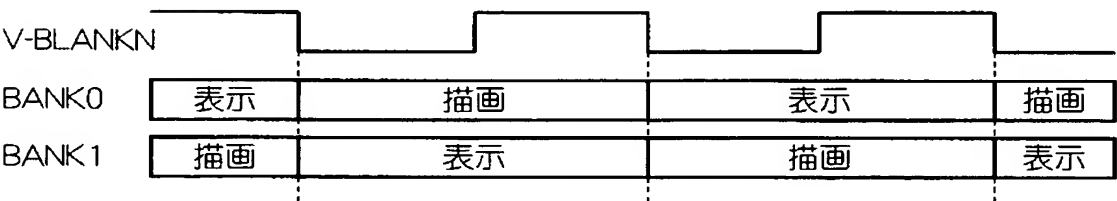
【図 2】



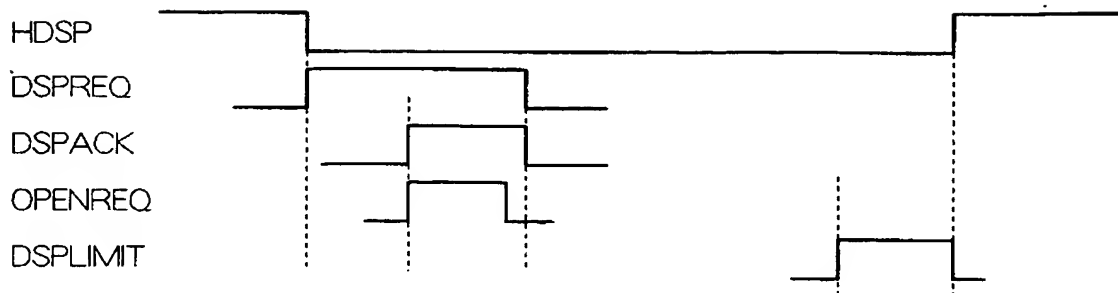
【図 3】



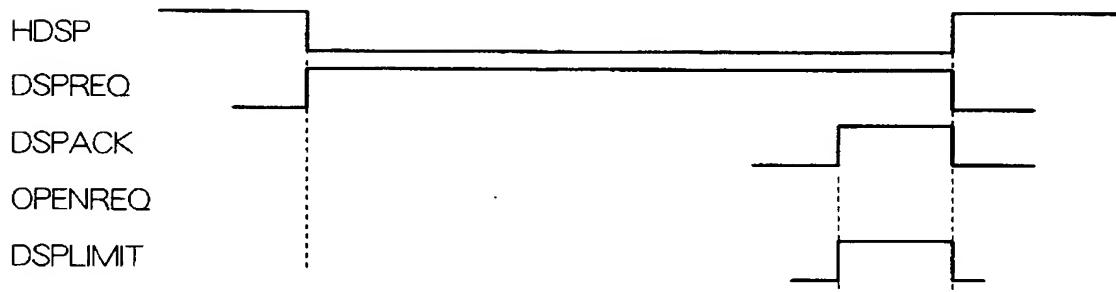
【図 4】



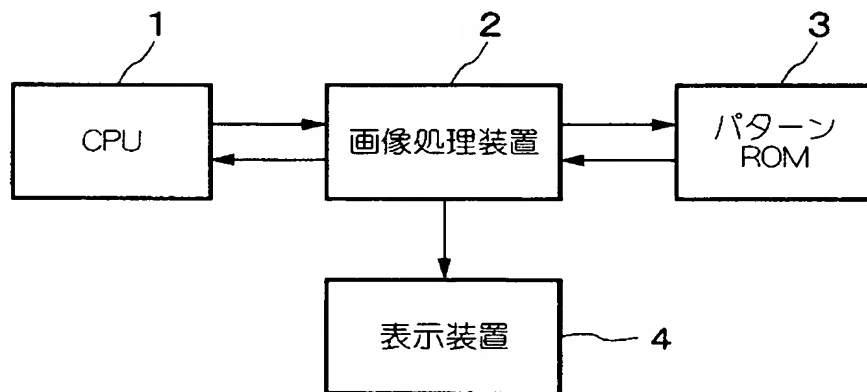
【図 5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 描画性能の向上を図った画像処理装置を提供する。

【解決手段】 デコーダ 16 は圧縮された画像データをデコードして圧縮前の画像データに戻す。インターフェイス 17 は、デコーダ 16 によってデコードされた画像データをスプライトバッファ 18 に書き込み、また、バッファ 18 内のデータを読み出し、レンダリングエンジン 21 へ出力する。レンダリングエンジン 21 は、バッファ 18 から読み出された画像データに所定のレンダリング処理を行ってフレームバッファ 23 に書き込む。ディスプレイコントローラ 25 はバッファ 23 から画像データを読み出し、表示装置 4 へ出力する。ここで、スプライトバッファ 18 はファーストイン・ファーストアウトメモリとして読出／書込が制御される。これにより、デコーダ 16 におけるデコードとレンダリングエンジン 21 における処理が非同期となり、描画性能を向上させることができる。

【選択図】 図 1

特願 2 0 0 3 - 0 2 3 4 4 4

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 4 0 7 5]

1. 変更年月日

1 9 9 0 年 8 月 2 2 日

[変更理由]

新規登録

住 所

静岡県浜松市中沢町 1 0 番 1 号

氏 名

ヤマハ株式会社